UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
09/183,335	10/30/98	FOSTER R	M-7085US

024251 WM02/0702
SKJERVEN MORRILL MACPHERSON LLP
25 METRO DRIVE
SUITE 700
SAN JOSE CA 95110

EXAMINER	
FULTS,R	
ART UNIT	**PAPER NUMBER**
2164	3

DATE MAILED: 07/02/01

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner of Patents and Trademarks

PTO-90C (Rev. 2/95)

1- File Copy

U.S. G.P.O. 2000 ; 465-188/25266

Office Action Summary	Application No.	Applicant(s)	
	09/183,335	FOSTER, ROBERT A.	
	Examiner	Art Unit	
	Richard Fults	2164	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136 (a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *30 October 1998* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-29* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-29* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claims _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are objected to by the Examiner.

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. § 119

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

Attachment(s)

15) ☒ Notice of References Cited (PTO-892)
16) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
17) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
18) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
19) ☐ Notice of Informal Patent Application (PTO-152)
20) ☐ Other: .

DETAILED ACTION

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Rejection, 35 U.S.C. 103(a)

Claims 1-5, 17-20, and 22-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gottesman et al (US 6,049,782 A) (hereinafter "Gottesman") and Petroutsos (Mastering Visual Basic 5), in view of each other as noted below.

As to **Claim 1** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, and col 8, lines 1-16) "A method for ....comprising: creating a plurality of product (component) rules....said financial transactions." , both as an inventive concept and as computer program functions to be performed, but he does not specifically teach the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program.

For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices to develop the computer logic and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate Gottesmans' invention with the teachings of Petroutsos because the combination of the two would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies. Likewise, in light of Petroutsos' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Petroutsos with those of Gottesman for the same reason.

As to **Claim 2** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, col 8, lines 1-16, and col 13) "The method of claim 1...comprising a billing method." , both as an inventive concept and as computer program functions to be performed, but he does not specifically teach the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed,

including billing statements, into a working computer software program. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices to develop the computer logic and the means for the pricing and other calculations required and to employ names, identifiers, and references for the databases and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed, including the billing method (statements). In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate Gottesmans' invention with the teachings of Petroutsos because the combination of the two would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies. Likewise, in light of Petroutsos' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Petroutsos with those of Gottesman for the same reason.

As to **Claim 3** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, col 8, lines 1-16, and col 13) "The method of Claim 1...display only information.", both as an inventive concept and as computer program functions to be performed, but he does not specifically teach the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like

Gottesman's described computer functions to be performed, into a working computer software program. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices to develop the computer logic and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed. It would also have been obvious that some of the specific names of the pricing and product (component) databases (tables) and fields would sometimes be utilized in a display mode only for purposes of identifying the information being viewed on the screen. Likewise, it would have been obvious that within the product (component) rules there would have resided information as to the status of the rule, as well as many other pieces of information as would have been needed to effectively operate the system. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate Gottesmans' invention with the teachings of Petroutsos because the combination of the two would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies. Likewise, in light of Petroutsos' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Petroutsos with those of Gottesman for the same reason.

As to **Claim 4** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, and col 8, lines 1-16) "The method of Claim 1....a price table.". , both as an inventive concept and as computer program functions to be performed, but he does not specifically teach the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays"

through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices to develop the computer logic and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate Gottesmans' invention with the teachings of Petroutsos because the combination of the two would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies. Likewise, in light of Petroutsos' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Petroutsos with those of Gottesman for the same reason.

As to **Claim 5** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, and col 8, lines 1-16) "The method of Claim 1....a pricing method." , both as an inventive concept and as computer program functions to be performed, but he does not specifically teach the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays"

through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices to develop the computer logic and the means for the many different method of pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate Gottesmans' invention with the teachings of Petroutsos because the combination of the two would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies. Likewise, in light of Petroutsos' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Petroutsos with those of Gottesman for the same reason.

As to **Claim 17** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, and col 8, lines 1-16) "The method of Claim 1...for said product rule". , both as an inventive concept and as computer program functions to be performed, but he does not specifically teach the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays"

through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices to develop the computer logic and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed. It is inherent in such programs to have both mandatory and optional attributes in their tables and rules. For example a table or field identifier is mandatory in order for the program to function properly, yet the number and type of fields and the individual rule descriptions and logic are discretionary. It would also be obvious to allow for the temporary disuse of one or more of those rules as an optional attribute, for a variety of operational reasons: possible temporary discontinuance of the product as one example, or a change in the attributes of the product itself as another reason. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate Gottesmans' invention with the teachings of Petroutsos because the combination of the two would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies. Likewise, in light of Petroutsos' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Petroutsos with those of Gottesman for the same reason.

As to **Claim 18** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, and col 8, lines 1-16) "The method of Claim 1....product rules to a database." , both as an inventive concept and as computer program functions to be performed, but he does not specifically teach the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices to develop the computer logic and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed. It was further obvious to one skilled in the art that testing computer software logic before implementation routinely requires 30% to 90% of the total time required to write any program of any complexity, and in this case that would include applying validating rules for validating the product rules. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate Gottesmans' invention with the teachings of Petroutsos because the combination of the two would have provided a completed, improved, and operational financial transaction system that could have been used by financial

service companies. Likewise, in light of Petroutsos' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Petroutsos with those of Gottesman for the same reason.

As to **Claim 19** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, and col 8, lines 1-16) "The method of Claim 1....a default product rule.", both as an inventive concept and as computer program functions to be performed, but he does not specifically teach the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices to develop the computer logic and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed. Whenever there are several choices of actions to be applied to a table of rules and one or more of the rules has no special action specified in the original design of the table, then it is obvious to provide a default action to that rule, and consequently provide for that eventuality in the design of the program logic. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the

invention to integrate Gottesmans' invention with the teachings of Petroutsos because the combination of the two would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies. Likewise, in light of Petroutsos' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Petroutsos with those of Gottesman for the same reason.

As to **Claim 20** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, and col 8, lines 1-16) "The method of Claim 1....price table contains prices.", both as an inventive concept and as computer program functions to be performed, but he does not specifically teach the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices to develop the computer logic and the means to create the prices within the price table and all the other calculations required in the program, and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the

time of the invention to integrate Gottesmans' invention with the teachings of Petroutsos because the combination of the two would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies. Likewise, in light of Petroutsos' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Petroutsos with those of Gottesman for the same reason.

As to **Claim 22** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, col 8, lines 1-16, col 10, lines 55-67, and cols 11-13) "The method of Claim 1, wherein said price table contains negative values", both as an inventive concept and as means for computer program functions to be performed, but he does not specifically teach the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices to develop the computer logic and the means and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed, including

specifically the use of negative values in the pricing tables. OFFICIAL NOTICE is taken that it would have been obvious for pricing tables to have contained negative values for the purpose of zeroing out a prior price entry by either multiplying it by a negative 1 or have the actual price be a negative number to be added to the prior entry in order to zero it out, to make the price table function properly. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate Gottesmans' invention and OFFICIAL NOTICE with the teachings of Petroutsos because the combination of the three would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies. Likewise, in light of Petroutsos' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Petroutsos and OFFICIAL NOTICE with those of Gottesman for the same reason.

As to **Claim 23** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, col 8, lines 1-16, col 10, lines 55-67, and cols 11-13) "A data processing system....said product rule and said price table.", both as an inventive concept and as means for computer program functions to be performed, but he does not specifically teach the detailed programatic steps for these software functions, such as "mandatory and optional" attributes as such. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. For such skilled

programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices to develop the computer logic and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed. It is inherent in such programs to always have both mandatory and optional attributes in their tables and rules. For example a table or field identifier is mandatory in order for the program to function properly, yet the number and type of fields and the individual rule descriptions and logic are discretionary. It would also be obvious to allow for the temporary disuse of one or more of those rules as an optional attribute, for a variety of operational reasons: possible temporary discontinuance of the product as one example, or a change in the attributes of the product itself as another reason. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate Gottesmans' invention with the teachings of Petroutsos because the combination of the two would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies. Likewise, in light of Petroutsos' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Petroutsos with those of Gottesman for the same reason.

As to **Claim 24** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, col 8, lines 1-16, col 10, lines 55-67, and cols 11-13) "The data processing system....means for billing.", both as an inventive concept and as means for computer program functions to be performed, but he does not specifically teach the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but

particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices to develop the computer logic and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed, including specifically the billing or statement preparation and delivery function. It is inherent in such programs to have both mandatory and optional attributes in their tables and rules. For example a table or field identifier is mandatory in order for the program to function properly, yet the number and type of fields and the individual rule descriptions and logic are discretionary. It would also be obvious to allow for the temporary disuse of one or more of those rules as an optional attribute, for a variety of operational reasons: possible temporary discontinuance of the product as one example, or a change in the attributes of the product itself as another reason. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate Gottesmans' invention with the teachings of Petroutsos because the combination of the two would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies. Likewise, in light of Petroutsos' teaching, it would

have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Petroutsos with those of Gottesman for the same reason.

As to **Claim 25** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, col 8, lines 1-16, col 10, lines 55-67, and cols 11-13) "The data processing system....display only information.", both as an inventive concept and as means for computer program functions to be performed, but he does not specifically teach the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices to develop the computer logic and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed, including specifically the means for creating the names of the product rules, the status of the product rules, how pricing and pricing is to be performed, and the means for creating display only information. It is inherent in such programs to always have both mandatory and optional attributes in their tables and rules. For example a table or field identifier is mandatory in order for the program to function properly, yet the number and type

of fields and the individual rule descriptions and logic are discretionary. It would also be obvious to allow for the temporary disuse of one or more of those rules as an optional attribute, for a variety of operational reasons: possible temporary discontinuance of the product as one example, or a change in the attributes of the product itself as another reason. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate Gottesmans' invention with the teachings of Petroutsos because the combination of the two would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies. Likewise, in light of Petroutsos' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Petroutsos with those of Gottesman for the same reason. See also the rejection reasons for Claim 3.

As to **Claim 26** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, col 8, lines 1-16, col 10, lines 55-67, and cols 11-13) "The data processing system of Claim 23....said mandatory attributes.", both as an inventive concept and as means for computer program functions to be performed, but he does not specifically teach the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for

multiple products (components) and multiple prices to develop the computer logic and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed, including specifically the billing or statement preparation and delivery function. It is inherent in such programs to always have both mandatory and optional attributes in their tables and rules. For example a table or field identifier is mandatory in order for the program to function properly, yet the number and type of fields and the individual rule descriptions and logic are discretionary. It would also be obvious to allow for the temporary disuse of one or more of those rules as an optional attribute, for a variety of operational reasons: possible temporary discontinuance of the product as one example, or a change in the attributes of the product itself as another reason. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate Gottesmans' invention with the teachings of Petroutsos because the combination of the two would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies. Likewise, in light of Petroutsos' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Petroutsos with those of Gottesman for the same reason.

As to **Claim 27** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, col 8, lines 1-16, col 10, lines 55-67, and cols 11-13) "The data processing system of Claim 26....said identifier.", both as an inventive concept and as means for computer program functions to be performed, but he does not specifically teach the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through

"Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices to develop the computer logic and the means and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed, including specifically the billing or statement preparation and delivery function. It is inherent in such programs to always have both mandatory and optional attributes in their tables and rules. For example a table or field identifier is mandatory in order for the program to function properly, yet the number and type of fields and the individual rule descriptions and logic are discretionary. It would also be obvious to allow for the temporary disuse of one or more of those rules as an optional attribute, for a variety of operational reasons: possible temporary discontinuance of the product as one example, or a change in the attributes of the product itself as another reason. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate Gottesmans' invention with the teachings of Petroutsos because the combination of the two would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies. Likewise, in light of Petroutsos' teaching, it would have been obvious to one

skilled in the art at the time of the invention to integrate the teachings of Petroutsos with those of Gottesman for the same reason.

As to **Claim 28** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, and col 8, lines 1-16) "The data processing system....rules to a database." , both as an inventive concept and as computer program functions to be performed, but he does not specifically teach the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices to develop the computer logic and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed. It was further obvious to one skilled in the art that testing computer software logic before implementation routinely requires 30% to 90% of the total time required to write any program of any complexity, and in this case that would include the means for validating the product rules. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate Gottesmans' invention with the teachings of Petroutsos because the combination

of the two would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies. Likewise, in light of Petroutsos' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Petroutsos with those of Gottesman for the same reason.

As to **Claim 29** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, col 8, lines 1-16, col 10, lines 55-67, and cols 11-13) "The data processing system of Claim 23....a default product rule.", both as an inventive concept and as means for computer program functions to be performed, but he does not specifically teach the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices to develop the computer logic and the means and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed, including specifically the billing or statement preparation and delivery function. It is inherent in such programs to always have both mandatory and optional attributes in their

tables and rules. For example a table or field identifier is mandatory in order for the program to function properly, yet the number and type of fields and the individual rule descriptions and logic are discretionary. It would also be obvious to allow for the temporary disuse of one or more of those rules as an optional attribute, for a variety of operational reasons: possible temporary discontinuance of the product as one example, or a change in the attributes of the product itself as another reason. Whenever there are several choices of actions to be applied to a table of rules and one or more of the rules has no special action specified in the original design of the table, then it is obvious to provide the means for a default action to that rule, and consequently provide for that eventuality in the design of the program logic. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate Gottesmans' invention with the teachings of Petroutsos because the combination of the two would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies. Likewise, in light of Petroutsos' teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Petroutsos with those of Gottesman for the same reason.

Rejection, 35 U.S.C. 103(a)

2. Claims 6-16 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gottesman et al (US 6,049,782 A) (hereinafter "Gottesman") and Petroutsos (Mastering Visual Basic 5), further in view of Carter (US 5,878,400) as noted below.

As to **Claim 6** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, col 8, lines 1-16, col 10, lines

55-67, and cols 11-13) "The method of Claim 5....is flat fee.", both as an inventive concept for pricing as commonly used in financial transactions and as means for computer pricing program functions to be performed, but he does not teach the many specific variations of pricing , all of which are old and well known, nor the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. Carter teaches (see at least Figure 7) approximately 25 different types of commonly used pricing types, which are only a partial listing of the many common pricing types, some of which have been used for the past several millenia. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices and pricing types to develop the computer logic for the many commonly known and used different types of pricing methods required for the many products involved and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed, including specifically flat fee pricing. OFFICIAL NOTICE is taken that one of the oldest and best known pricing types is the flat fee type, which has long been commonly used for services of all types, such as Doctors, gardeners, hairdressers, and bank services, etc., and that it would have been obvious to modify the teachings of Carter with this OFFICIAL

NOTICE. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to employ the teachings of Gottesman to the teachings of Petroutsos as modified by Carter because the combination of the three would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies, complete with a wide variety of pricing types. Likewise, in light of Carter's teaching and that of OFFICIAL NOTICE, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Carter and OFFICIAL NOTICE and Petroutsos with those of Gottesman for the same reason.

As to **Claim 7** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, col 8, lines 1-16, col 10, lines 55-67, and cols 11-13) "The method of Claim 5....is unit price.", both as an inventive concept for pricing as commonly used in financial transactions and as means for computer pricing program functions to be performed, but he does not teach the many specific variations of pricing , all of which are old and well known, nor the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. Carter teaches (see at least Figure 7) approximately 25 different types of commonly used pricing types, which are only a partial listing of the many common pricing types, some of which have been used for the past several millenia. For such skilled programmers it would have been obvious in a

financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices and pricing types to develop the computer logic for the many commonly known and used different types of pricing methods required for the many products involved and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed, including specifically unit price pricing. OFFICIAL NOTICE is taken that one of the oldest and best known pricing types is the unit price type, which has long been commonly used for goods of all types, such as groceries, office supplies, clothes, and bank services, etc., and that it would have been obvious to modify the teachings of Carter with this OFFICIAL NOTICE. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to employ the teachings of Gottesman to the teachings of Petroutsos as modified by Carter because the combination of the three would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies, complete with a wide variety of pricing types. Likewise, in light of Carter's teaching and that of OFFICIAL NOTICE, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Carter and OFFICIAL NOTICE and Petroutsos with those of Gottesman for the same reason.

As to **Claim 8** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, col 8, lines 1-16, col 10, lines 55-67, and cols 11-13) "The method of Claim 5....is unit cost.", both as an inventive concept for pricing as commonly used in financial transactions and as means for computer pricing program functions to be performed, but he does not teach the many specific variations of pricing , all of which are old and well known,

nor the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. Carter teaches (see at least Figure 7) approximately 25 different types of commonly used pricing types, including base cost, which are only a partial listing of the many common pricing types, some of which have been used for the past several millenia. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices and pricing types to develop the computer logic for the many commonly known and used different types of pricing methods required for the many products involved and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed, including specifically unit cost pricing. OFFICIAL NOTICE is taken that one of the oldest and best known pricing types is the unit cost type, which has long been commonly used for goods and services in special circumstances, such as for sales of goods or services which were in addition to other sales being made at the same time to the same customer or as an special incentive or a promotional price, and that it would have been obvious to modify the teachings of Carter with this OFFICIAL NOTICE. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to employ the

teachings of Gottesman to the teachings of Petroutsos as modified by Carter because the combination of the three would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies, complete with a wide variety of pricing types. Likewise, in light of Carter's teaching and that of OFFICIAL NOTICE, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Carter and OFFICIAL NOTICE and Petroutsos with those of Gottesman for the same reason.

As to **Claim 9** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, col 8, lines 1-16, col 10, lines 55-67, and cols 11-13) "The method of Claim 5....is volume discount.", both as an inventive concept for pricing as commonly used in financial transactions and as means for computer pricing program functions to be performed, but he does not teach the many specific variations of pricing , all of which are old and well known, nor the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. Carter teaches (see at least Figure 7) approximately 25 different types of commonly used pricing types, including volume discount, which are only a partial listing of the many common pricing types, some of which have been used for the past several millenia. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products

(components) and multiple prices and pricing types to develop the computer logic for the many commonly known and used different types of pricing methods required for the many products involved and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed, including specifically volume discount pricing. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to employ the teachings of Gottesman to the teachings of Petroutsos as modified by Carter because the combination of the three would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies, complete with a wide variety of pricing types. Likewise, in light of Carter's teaching it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Carter and Petroutsos with those of Gottesman for the same reason.

As to **Claim 10** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, col 8, lines 1-16, col 10, lines 55-67, and cols 11-13) "The method of Claim 5....is tiering.", both as an inventive concept for pricing as commonly used in financial transactions and as means for computer pricing program functions to be performed, but he does not teach the many specific variations of pricing , all of which are old and well known, nor the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to

create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. Carter teaches (see at least Figure 7) approximately 25 different types of commonly used pricing types, including tiering, which are only a partial listing of the many common pricing types, some of which have been used for the past several millenia. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices and pricing types to develop the computer logic for the many commonly known and used different types of pricing methods required for the many products involved and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed, including specifically tiering pricing. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to employ the teachings of Gottesman to the teachings of Petroutsos as modified by Carter because the combination of the three would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies, complete with a wide variety of pricing types. Likewise, in light of Carter's teaching it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Carter and Petroutsos with those of Gottesman for the same reason.

As to **Claim 11** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, col 8, lines 1-16, col 10, lines 55-67, and cols 11-13) "The method of Claim 5....is cost plus.", both as an inventive concept for pricing as commonly used in financial transactions and as means for computer pricing program functions to be performed, but he does not

teach the many specific variations of pricing , all of which are old and well known, nor the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. Carter teaches (see at least Figure 7) approximately 25 different types of commonly used pricing types, including cost plus, which are only a partial listing of the many common pricing types, some of which have been used for the past several millenia. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices and pricing types to develop the computer logic for the many commonly known and used different types of pricing methods required for the many products involved and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed, including specifically cost plus pricing. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to employ the teachings of Gottesman to the teachings of Petroutsos as modified by Carter because the combination of the three would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies, complete with a wide variety of pricing types. Likewise, in light of Carter's teaching it would have been obvious to one

skilled in the art at the time of the invention to integrate the teachings of Carter and Petroutsos with those of Gottesman for the same reason.

As to **Claim 12** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, col 8, lines 1-16, col 10, lines 55-67, and cols 11-13) "The method of Claim 5....is minimum revenue.", both as an inventive concept for pricing as commonly used in financial transactions and as means for computer pricing program functions to be performed, but he does not teach the many specific variations of pricing , all of which are old and well known, nor the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. Carter teaches (see at least Figure 7) approximately 25 different types of commonly used pricing types, some of which are only a partial listing of the many common pricing types, some of which have been used for the past several millenia. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices and pricing types to develop the computer logic for the many commonly known and used different types of pricing methods required for the many products involved and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all

together for the purpose and function intended to be performed, including specifically minimum revenue pricing. OFFICIAL NOTICE is taken that minimum revenue pricing is old and well known, and has long been commonly used for goods and services in special price calculation circumstances, such as whenever the proscribed calculation method had yielded only a very small nominal value that may not have included overhead or transactional expenses and the management position had been taken that all such transactions would have had at least a preset minimum revenue as its charged price, and that it would have been obvious to modify the teachings of Carter with this OFFICIAL NOTICE. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to employ the teachings of Gottesman to the teachings of Petroutsos as modified by Carter and OFFICIAL NOTICE because the combination of the four would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies, complete with a wide variety of pricing types. Likewise, in light of Carter's teaching and that of OFFICIAL NOTICE, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Carter and OFFICIAL NOTICE and Petroutsos with those of Gottesman for the same reason.

As to **Claim 13** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, col 8, lines 1-16, col 10, lines 55-67, and cols 11-13) "The method of Claim 5....is maximum revenue.", both as an inventive concept for pricing as commonly used in financial transactions and as means for computer pricing program functions to be performed, but he does not teach the many specific variations of pricing , all of which are old and well known, nor the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management

Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. Carter teaches (see at least Figure 7) approximately 25 different types of commonly used pricing types, some of which are only a partial listing of the many common pricing types, some of which have been used for the past several milenia. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices and pricing types to develop the computer logic for the many commonly known and used different types of pricing methods required for the many products involved and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed, including specifically maximum revenue pricing. OFFICIAL NOTICE is taken that maximum revenue pricing is old and well known, and has long been commonly used for goods and services in special price calculation circumstances, such as whenever the proscribed calculation method had yielded an unusually high value that may not have been competitive or not have appeared reasonable to the customer, and the management position had been taken that all such transactions will have had at most a preset maximum revenue as its charged price, and that it would have been obvious to modify the teachings of Carter with this OFFICIAL NOTICE. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to employ the teachings of Gottesman to the teachings of Petroutsos as modified by Carter and OFFICIAL NOTICE because the combination of the FOUR would have provided a

completed, improved, and operational financial transaction system that could have been used by financial service companies, complete with a wide variety of pricing types. Likewise, in light of Carter's teaching and that of OFFICIAL NOTICE, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Carter and OFFICIAL NOTICE and Petroutsos with those of Gottesman for the same reason.

As to **Claim 14** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, col 8, lines 1-16, col 10, lines 55-67, and cols 11-13) "The method of Claim 5....is markup of total price.", both as an inventive concept for pricing as commonly used in financial transactions and as means for computer pricing program functions to be performed, but he does not teach the many specific variations of pricing , all of which are old and well known, nor the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. Carter teaches (see at least Figure 7) approximately 25 different types of commonly used pricing types, some of which are only a partial listing of the many common pricing types, some of which have been used for the past several milenia. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices and pricing types to develop the computer logic for the many commonly known

and used different types of pricing methods required for the many products involved and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed, including specifically markup of total price pricing. OFFICIAL NOTICE is taken that markup of total price as a pricing method is old and well known, and has long been commonly used for goods and services in sales tax calculations and in special price calculation circumstances, as one example such as whenever the proscribed calculation method became outdated due to a price increase not yet there reflected, which required that an additional markup be charged on top of the total price, and that it would have been obvious to modify the teachings of Carter with this OFFICIAL NOTICE. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to employ the teachings of Gottesman to the teachings of Petroutsos as modified by Carter and OFFICIAL NOTICE because the combination of the four would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies, complete with a wide variety of pricing types. Likewise, in light of Carter's teaching and that of OFFICIAL NOTICE, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Carter and OFFICIAL NOTICE and Petroutsos with those of Gottesman for the same reason.

As to **Claim 15** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, col 8, lines 1-16, col 10, lines 55-67, and cols 11-13) "The method of Claim 5....is bundled pricing.", both as an inventive concept for pricing as commonly used in financial transactions and as means for computer pricing program functions to be performed, but he does not teach the many specific variations of pricing , all of which are old and well known,

nor the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. Carter teaches (see at least Figure 7) approximately 25 different types of commonly used pricing types, some of which are only a partial listing of the many common pricing types, some of which have been used for the past several milenia. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices and pricing types to develop the computer logic for the many commonly known and used different types of pricing methods required for the many products involved and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed, including specifically bundled pricing. OFFICIAL NOTICE is taken that one of the oldest and best known pricing types is the bundled pricing type, which has long been commonly used for goods as part of the normal pricing method, as one example the sale of new cars, wherein several pieces of optional equipment that have been individually priced at full retail were routinely included with the car at a discounted value (bundled pricing) when purchased all together with the car, and that it would have been obvious to modify the teachings of Carter with this OFFICIAL NOTICE. In view of Gottesmans' teaching, it would have been obvious

to one skilled in the art at the time of the invention to employ the teachings of Gottesman to the teachings of Petroutsos as modified by Carter and OFFICIAL NOTICE because the combination of the four would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies, complete with a wide variety of pricing types. Likewise, in light of Carter's teaching and that of OFFICIAL NOTICE, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Carter and OFFICIAL NOTICE and Petroutsos with those of Gottesman for the same reason.

As to **Claim 16** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 6, lines 25-67, col 7, lines 48-67, col 8, col 9, col 10, and cols 11-13) "The method of Claim 5....is Cross CAA Bundled Tiering.", which he describes as Relationship Pricing, both as an inventive concept for pricing as commonly used in financial transactions and as means for computer pricing program functions to be performed, and although he does teach tiering, he does not teach the many specific variations of pricing by name, all of which are old and well known, nor the detailed programatic steps for these software functions. However, it is inherent in financial institution pricing to analyze customer accounts, both across their several accounts and across the acounts of all major customers, and to employ a wide variety of pricing methods to price their services, including bundling and tiering, especially in relationship pricing which is essentially bundled pricing by definition. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's

described computer functions to be performed, into a working computer software program. Carter teaches (see at least Figure 7) approximately 25 different types of commonly used pricing types , including 5 which contain the word "customer", which are only a partial listing of the many common pricing types, some of which have been used for the past several milenia. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices and pricing types to develop the computer logic for the many commonly known and used different types of pricing methods required for the many products involved and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed, including specifically cross customer account analysis (CAA) bundled tiering pricing, which is inherent in and is relationship pricing. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to employ the teachings of Gottesman to the teachings of Petroutsos as modified by Carter because the combination of the three would have provided a completed, improved, and operational financial transaction system that could have been used by financial service companies, complete with a wide variety of pricing types. Likewise, in light of Carter's teaching, it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Carter and Petroutsos with those of Gottesman for the same reason.

As to **Claim 21** Gottesman discloses (see at least cols 1-14, but particularly col 4, lines 42-67, col 5, lines 1-57, col 7, lines 48-67, col 8, lines 1-16, col 10, lines 55-67, and cols 11-13) "The method of Claim 1, wherein said price table contains costs.", both as an inventive concept for pricing as commonly used in financial transactions and as means for computer pricing program functions to be

performed, but he does not teach the many specific variations of pricing , all of which are old and well known, nor the detailed programatic steps for these software functions. Petroutsos teaches (see his book in general, but particularly Chapter 3 at least the sections "Arrays" through "Arrays of Arrays", and "If...Then...End", and Chapter 11 at least the sections "Databases and Database Management Systems" through "The Data Control's Properties"), by example of one of several computer programming languages then available, how relatively logical, simple, and obvious it would have been for one skilled in the art at the time of the invention to create the programming steps necessary to translate an inventive concept, like Gottesman's described computer functions to be performed, into a working computer software program. Carter teaches (see at least Figure 7) approximately 25 different types of commonly used pricing types, including base cost and one he terms "give it to them for cost", which are only a partial listing of the many common pricing types, some of which have been used for the past several millenia. The at-cost prices also serve as cost information outside of the cost accounting system for any person who wishes to review it. For such skilled programmers it would have been obvious in a financial transaction system containing price tables and product (component) rules for multiple products (components) and multiple prices and pricing types to develop the computer logic for the many commonly known and used different types of pricing methods required for the many products involved and the means for the pricing and other calculations required and to employ specific names, identifiers, and references for the pricing and product (component) databases (tables) and their data fields, together with the programatic means to both create all of the logic and tables and to link them all together for the purpose and function intended to be performed, including specifically product costs. In view of Gottesmans' teaching, it would have been obvious to one skilled in the art at the time of the invention to employ the teachings of Gottesman to the teachings of Petroutsos as modified by Carter because the combination of the three would have provided a completed, improved, and operational financial transaction system that could

have been used by financial service companies, complete with a wide variety of pricing types. Likewise, in light of Carter's teaching it would have been obvious to one skilled in the art at the time of the invention to integrate the teachings of Carter and Petroutsos with those of Gottesman for the same reason.

3. The prior art of record, although not cited above, is considered pertinent to one or more of the Applicants' claimed inventions:

US 4,855,908 to Skimoda et al, which teaches price table lookups.

US 5,710,887 to Chelliah et al, which teaches pricing rules.

Boris Beizer, Software Testing Techniques, International Thomson Computer Press, Boston, MA, 1990, which teaches the need to test software before implementation.

Paul Carrubba, Principles of Banking, American Banking Association, 1994, which teaches that pricing is based upon account analysis.

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Richard Fults whose telephone number is 703-305-5416. The examiner can normally be reached on Every day 8:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Vincent Millin can be reached on 703-308-1065. The fax phone numbers for the organization where this application or proceeding is assigned are 703-308-1396 for regular communications and 703-308-1396 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 703-305-3900.

6/26/01

MELANIE ACKEMPER
PRIMARY EXAMINER